



Git Commit Message Convention

☰ 태그

♥ 커밋 메시지 컨벤션



1. 커밋 유형 지정

- 커밋 유형은 영어 대문자로 작성하기

커밋 유형	의미
Feat	새로운 기능, 파일 추가
Fix	버그 수정
Docs	문서 수정
Style	코드 formatting, 세미콜론 누락, 코드 자체의 변경이 없는 경우
Refactor	코드 리팩토링
Test	테스트 코드, 리팩토링 테스트 코드 추가
Chore	패키지 매니저 수정, 그 외 기타 수정 ex) .gitignore
Design	CSS 등 사용자 UI 디자인 변경
Comment	필요한 주석 추가 및 변경
Rename	파일 또는 폴더 명을 수정하거나 옮기는 작업만인 경우
Remove	파일을 삭제하는 작업만 수행한 경우
!BREAKING CHANGE	커다란 API 변경의 경우
!HOTFIX	급하게 치명적인 버그를 고쳐야 하는 경우

2. 제목과 본문을 빈행으로 분리

- 커밋 유형 이후 제목과 본문은 한글로 작성하여 내용이 잘 전달될 수 있도록 할 것
- 본문에는 변경한 내용과 이유 설명 (어떻게보다는 무엇 & 왜를 설명)

3. 제목 첫 글자는 대문자로, 끝에는 . 금지

4. 제목은 영문 기준 50자 이내로 할 것

5. 자신의 코드가 직관적으로 바로 파악할 수 있다고 생각하지 말자

6. 여러가지 항목이 있다면 글머리 기호를 통해 가독성 높이기

- 변경 내용 1
- 변경 내용 2
- 변경 내용 3

♥ 규칙에 맞는 좋은 커밋메시지를 작성해야 하는 이유

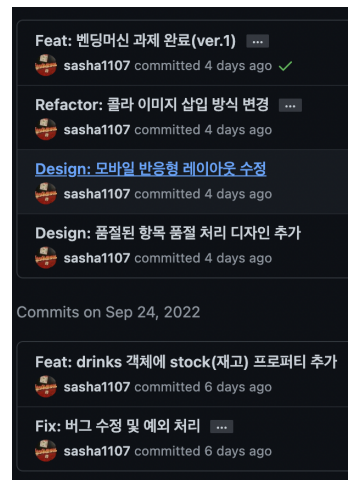
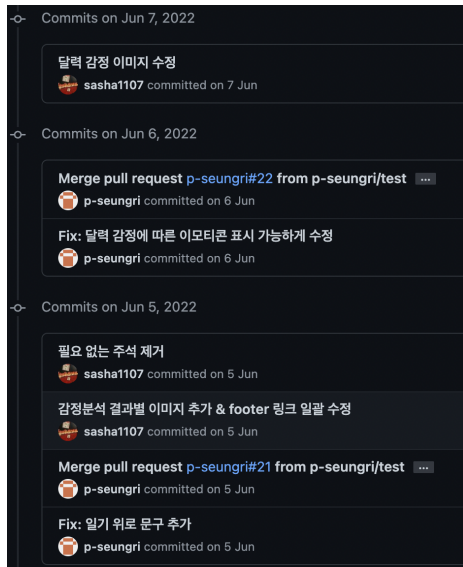
- 팀원과의 소통
- 편리하게 과거 추적 가능
- 나중에 실무에서 익숙해지기 위해



안지킨 사례



지킨 사례



♥ 한 커밋에는 한 가지 문제만!

- 추적 가능하게 유지해주기
- 너무 많은 문제를 한 커밋에 담으면 추적하기 어렵다.

♥ CLI에서 커밋 메시지 여러 줄로 작성하는 방법



쌍따옴표를 닫지 말고 개행하며 작성 → 다 작성한 후에 쌍따옴표를 닫으면 작성 완료

```
git commit -m "FEAT: 회원가입 기능 추가
- 회원가입 기능 추가"
```

```
* soohyun ~/Documents/TIL main + git commit -m "1만 시간의 법칙 완성 ver 1.0
dquote>
dquote> - flex, grid 사용하지 않고 position으로 요소 배치
dquote> - 모달창 구현 X
dquote> - 모바일 버전 구현 X"
```

브랜치 전략

- master
- develop
 - feature들을 머징 하는 브랜치
- feature/[back or front]-[영문 기능]
 - 구현해야하는 기능들을 개발하는 브랜치

ex) feature/front-login-page